

Questions de cours

1. Quelle différence faites-vous entre un microcontrôleur, microprocesseur et un SOC ?

Un microcontrôleur intègre toutes les ressources nécessaires au fonctionnement d'un système numérique (CPU, RAM, ROM, E/S, CAN, timers, ...) avec des ressources limitées. En général, il ne supporte pas de système d'exploitation, sa mise en œuvre électronique est simple, les programmes sont prédictibles et il est peu cher mais il dispose de peu de ressources.

Un microprocesseur doit être associé à d'autres composants (RAM, ROM, contrôleurs d'entrées/sorties) pour réaliser un système numérique. Il dispose de beaucoup plus de puissance de calcul (multi-core, architecture 32 ou 64 bits), il est fait pour supporter un système d'exploitation, ce qui permet de développer des programmes sans avoir à gérer le matériel (drivers) et dispose d'une unité de gestion de mémoire virtuelle. Il permet d'exécuter plusieurs tâches simultanément. Sa mise en œuvre matérielle est plus complexe et donc plus coûteuse.

Un soc (System On Chip) est un circuit intégré dans lequel on trouve un microprocesseur, de la ROM et des contrôleurs d'entrées/sorties. Il cumule les avantages des microcontrôleurs (intégration matérielle) et des microprocesseurs (puissance de calcul).

2. Qu'est-ce que l'ordonnancement ? Quels sont les deux principaux modes d'ordonnements des systèmes d'exploitation ?

L'ordonnancement est la tâche de sélection d'un processus en attente dans la liste des processus prêts et d'allocation du CPU pour ce processus.

Les deux principaux modes d'ordonnements des systèmes d'exploitation sont le temps partagé et le temps réel.

3. Quel est le mode d'ordonnement par défaut de Linux ?

Temps partagé mais depuis la version 2.6.23 : temps réel souple (ordonnement CFS) sans garanties.

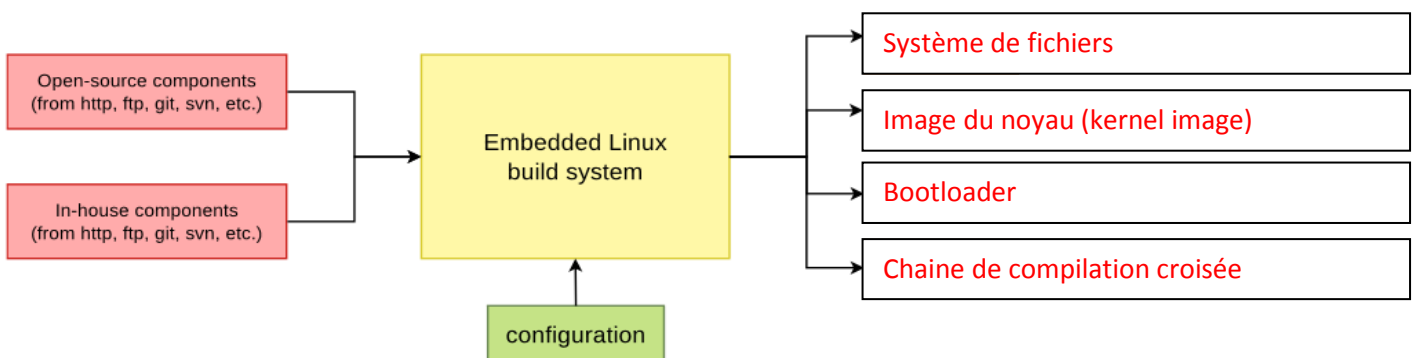
4. Que permet d'obtenir l'application du patch PREEMPT_RT sur le noyau Linux ?

L'application du patch PREEMPT_RT au noyau Linux lui confère des propriétés temps réel souple améliorées (centaines de microsecondes et comportement prévu pour gérer les pires circonstances)

5. Qu'est-ce qu'un build system ? Donnez le nom du build system que vous avez utilisé en TP.

Un build system est un ensemble de logiciels permettant de construire un système d'exploitation et des outils de développement sur mesure pour une cible donnée. En TP, on a utilisé Buildroot.

6. Quels sont les éléments construits par le build system. Complétez le schéma ci-dessous :



7. Qu'est-ce que la MMU ? Quelle différence y-a-t-il entre une adresse virtuelle et une adresse physique ?

La MMU est l'unité de gestion de la mémoire. C'est un dispositif intégré au microprocesseur qui assure la conversion des adresses virtuelles, générées par le CPU et utilisées par les programmes utilisateur en adresses physiques, vues par l'unité de mémorisation, incluant la mémoire centrale et secondaire.

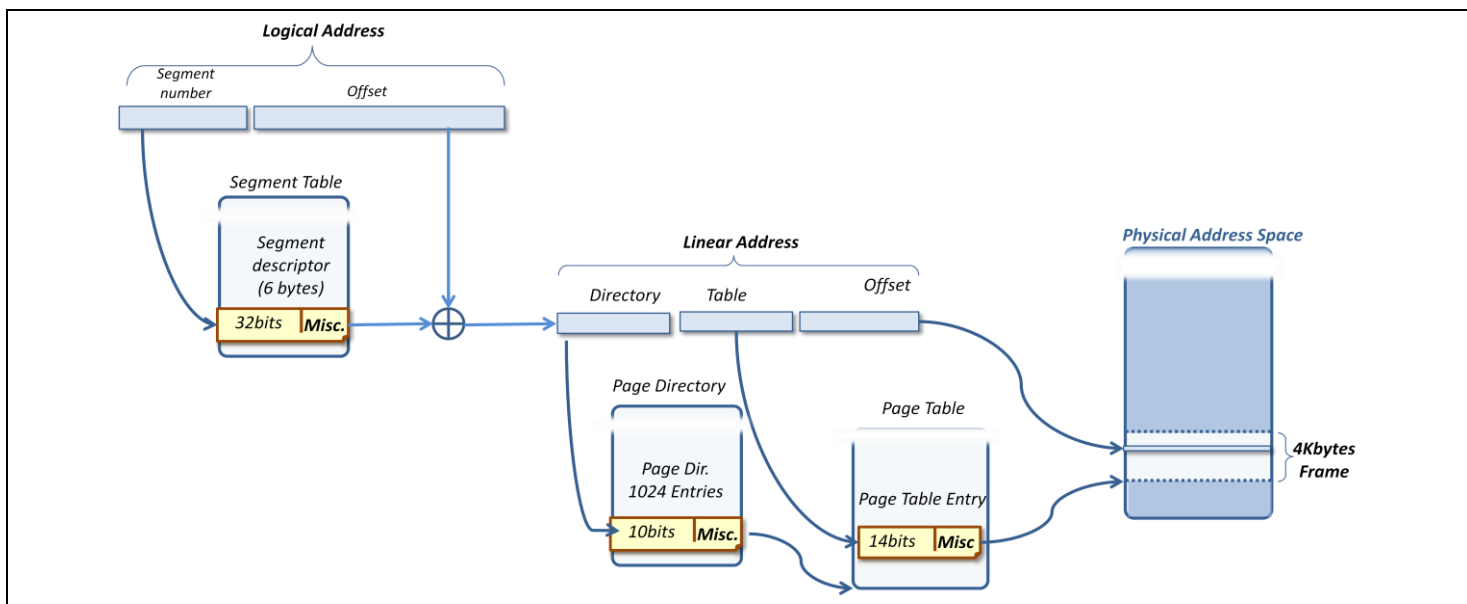
8. Qu'est-ce que le TLB ? Dans quel mécanisme de gestion mémoire est-il utilisé ?

Le TLB (Translation Lookaside Buffer) est une mémoire cache additionnelle qui contient les traductions des adresses virtuelles récrémentées utilisées en adresses physiques. Il est utilisé en parallèle au mécanisme de pagination et permet d'accélérer la traduction de ses adresses.

Gestion de mémoire dans un système d'exploitation

Soit un système de gestion de mémoire gérée de manière segmentée et paginée avec double niveau de pagination.

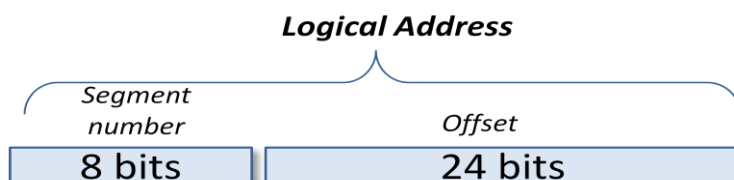
- La taille de la mémoire physique est de 64 Mo (1mot = 1 octet).
- Un processus peut avoir au plus 256 segments.
- Chaque segment peut adresser au plus 16 Mo.
- La taille d'une page est fixée à 4 ko.



1. Quel est le format des adresses logiques ? Expliquez.

Une adresse logique est composée d'un n° de segment et d'un décalage.

- Le numéro de segment doit pouvoir représenter 256 segments => 8 bits
- Le décalage doit pouvoir adresser un espace de 16 Mo => 24 bits



2. Quel est le format d'une adresse physique ? Expliquez.

L'espace d'adressage physique est de 64 Mo => 26 bits.

3. Quelle est la taille de l'espace d'adressage virtuelle

1 adresse virtuelle = 32 bits donc espace d'adressage de 4 Go ou 256 segments de 16 Mo au max = 32 Go.

Soit un processus muni de la table des segments suivante :

Segment	Base	Limite
00	00 BE 0A 00	10 00
01	00 BE 23 D1	02 FF
02	00 BE 00 DA	03 61
03	00 BE 1A 26	05 07
04	00 BE 0F F0	10 00

du répertoire de pages suivant :

Répertoire	Table	Active
0	0	1
1	3	0
2	1	1
3	2	0

Et de deux tables de pages :

N°	@Page	@Cadre	Active	Libre
0	2A0	23 40	1	0
1	2A1	05 BB	1	0
2	2A2	00 00	0	0
3	2A3	14 E0	1	0
4				1
				1
1023				1

Table 0

N°	@Page	@Cadre	Active	Libre
0	3E0	00 00	0	0
1	3E1	27 FD	1	0
2	3E2	00 00	0	0
3	3E3	3A F6	1	0
4				1
				1
1023				1

Table 1

- Quelle est l'adresse linéaire correspondante à l'adresse logique **0x03000F0**
 - N° de segment : 3 => adresse de base : **0X00BE1A26**
 - Décalage : **0x000F0** bien inférieur à la limite du segment (0x0507) pas d'erreur de dépassement
 - Adresse linéaire = **0x00BE1A26 + 0x000F0 = 0x00BE1B16**
- Quel est le format d'une adresse linéaire ? Expliquez.
 - Le répertoire de page contient 1024 entrées => 10 bits pour identifier une entrée
 - Une page contient 1024 entrées => 10 bits pour identifier une entrée
 - La taille d'une page est de 4 ko => 12 bits de décalage



- Quelle est l'adresse physique correspondante à l'adresse logique **0x0300F000**

Adresse logique **0x0300F000** <=> Adresse linéaire **0X00BE1B16**

0	0	0	0	0	0	0	0	0	1	0	1	1	1	1	1	0	0	0	0	1	1	0	1	1	0	0	0	1	0	1	1	0
Répertoire : 2										Table : 3E1										Décalage : B16												

- L'entrée 2 du répertoire de page indique la table de pages 1.
 - L'entrée **0x3E1** de cette table indique le cadre **0x27FD**, qui correspond aux 14 bits de poids fort de l'adresse physique
 - Le décalage est égal à **0xB16**, qui correspond aux 12 bits de poids faible de l'adresse physique
 - L'adresse physique correspondante : **0x27FDB16**
- Quelle quantité de mémoire physique occupe le processus.
 - 1 répertoire de table contenant 4 numéros de table (de 0 à 3) et leur état d'activité (0 ou 1).
 - Numéro de table codé sur 10 bits et état d'activité sur 1 bit. Ces informations nécessitent 2 octets.
 - Quantité de mémoire physique occupée par le répertoire de table : $4 \times 2 = 8$ octets (**négligeable**)
 - 2 tables contenant 1024 entrées chacune.
 - Numéro de cadre codé sur 14 bits + 1 bit d'activité + 1 bit de liberté : Nécessite 2 octets
 - Quantité de mémoire physique occupée par les tables : $2 \times 2 \times 1024 = 4$ ko
 - 5 pages actives et une page occupe 4 ko : $5 \times 4 = 20$ ko
 - Occupation totale en mémoire physique : **24 ko**